



# AI-Based Stock Prediction, Analysis, and Chatbot Platform



## Overview

This project integrates **deep learning**, **agentic AI**, and **generative AI** to build a smart, web-based stock analysis and trading assistant. It empowers users with:

- **Accurate future stock predictions**
- **Real-time news and market updates**
- **AI chatbot for instant user support**
- **Buy/Sell functionality with intuitive charts**

Technologies used include **FastAPI**, **React.js**, **LangChain**, **Groq API**, **YFinance**, and advanced ML models (LSTM, GRU, etc.).

---



## Architecture

### ◆ Frontend

- **React.js** with Chart.js for data visualization
- Material-UI/Bootstrap for UI components

### ◆ Backend

- **FastAPI** (REST API) to handle:
  - Stock predictions
  - Chatbot interaction
  - Real-time market data
  - User trading requests

### ◆ AI Integration

- Deep Learning Models (LSTM, GRU, Transformer)
  - Agentic AI using Phi Framework + Groq
  - LangChain with Groq/Gemini/Ollama for chatbot
-

# Technologies & Components

## 1. Stock Price Prediction (Deep Learning)

### Models Used:

- LSTM: Good for sequential trends
- GRU: Lighter alternative to LSTM
- Transformer: Excellent for long sequences and faster training
- **Recommended:** Informer or Temporal Fusion Transformer (TFT) for advanced time-series forecasting

### Libraries:

TensorFlow, Keras, PyTorch, scikit-learn, pandas, numpy

## 2. Agentic AI for Market & News Updates

### Stack Used:

- **Groq API:** Ultra-fast LLM for real-time responses
- **YFinanceTools:** Pulls latest stock market data
- **DuckDuckGo:** Gathers global news & sentiments

**Output:** Aggregated stock analysis and latest news

## 3. Generative AI Chatbot

### Tools Used:





- LangChain with support for:
  - Groq (blazing-fast inference)
  - Gemini (Google's multi-modal LLM)
  - Ollama (run local LLMs)

### Recommendation:

Use **LangChain + Groq** for production due to speed. Add Gemini for multi-modal or web data responses.

---

# Features

-  **Stock Predictions** (using deep learning)
-  **News Updates** (agent-based via DuckDuckGo)
-  **Customer Chatbot** (LangChain + Groq)
-  **Buy/Sell Stocks** (basic simulated trading API)

-  **Real-Time Charts** (React + Chart.js)
- 



## Bonus/Advanced Ideas

- **Sentiment Analysis** from tweets/news headlines using HuggingFace DistilBERT
  - **Auto Portfolio Rebalancing** using RL
  - **Voice Assistant Integration** with Web Speech API
  - **User Authentication** (OAuth2, JWT tokens)
  - **User Trading Analytics Dashboard**
  - **TradingView API** or **Alpaca API** integration for real-time trades
- 



## Folder Structure (Suggestion)

```
project-root/
├── backend/                                # FastAPI backend
│   ├── main.py
│   ├── agents/
│   ├── models/                            # LSTM, GRU, etc.
│   ├── routes/
│   └── utils/
├── frontend/                             # React.js frontend
│   ├── public/
│   └── src/
│       ├── components/
│       ├── pages/
│       └── services/
├── requirements.txt
└── README.md
```

---



## Resources

- [Groq API](#)
  - [Phi Agents GitHub](#)
  - [LangChain Docs](#)
  - [YFinance](#)
  - [TradingView Widgets](#)
-